

NETWORK LINK DETECTION AND GENERATION

This application is a continuation-in-part of U.S. Ser. No. 07/971,018, filed Nov. 2, 1992, abandoned for "Network Link Endpoint Capability Detection," incorporated herein by reference. 5

The present invention is directed to a method and apparatus for generating and detecting, in a network, such as a local area network, the link signals transmitted to or received from one or more endpoints of a data communication link, and in particular to a method and apparatus for generating one or a plurality of different link signals and determining whether a data source/sink at the end of a datalink has the capability of first data communication protocol or a second data communication protocol. 15

BACKGROUND OF THE INVENTION

A typical data communication network is configured to operate according to a single predetermined protocol, e.g., an Ethernet protocol, a token ring protocol, other LAN protocols, or an isochronous protocol. An example of an Ethernet system is an implementation known as 10 Base T which is described in the draft Nine supplement to IEEE standard 802.3, dated Nov. 15, 1989. Other examples of data communication protocols are X.25, and the Token Ring System, described for example, by IEEE Standard 802.5. Both Ethernet and token ring systems convey data in packets but each uses a different media access method. 20

As shown in FIG. 1A, in a packet system, data is transferred in a plurality of packets 12a, 12b which can be either constant-sized or variable-sized. Each packet includes a field of data 14a, 14b which may be preamble information 16a, 16b housekeeping information such as data source information, data destination information, and the like 18a, 18b and a frame end marker 20a. As seen in FIG. 1A, because the fields provided for data 14a, 14b are not substantially continuous, the packetized scheme of FIG. 1A is not isochronous but "bursty" in nature. 25

In a token ring system, a node is permitted to transmit data only after receipt of an electronic "token." As depicted in FIG. 1B, a first station may transmit a token 22 which is received 24a by second station whereupon the second station may begin transmission of data 26a. After a period of data transmission, the second station transmits the token 22b which is received by a third station 24b that can then begin its own transmission of data 26b. As seen in FIG. 1B, because data transmission is synchronized with the occurrence of an event (the arrival of a token), the token ring system is not an isochronous data transfer system. 30

FIG. 1C schematically depicts isochronous data transfer. In general terms, isochronous data is data which is non-packetized and of indeterminate, potentially continuous duration. In isochronous data transfer, the data transfer is initiated, such as by initiating a telephone conversation or beginning a video camera transmission 30. After the data transfer or connection is initiated, transmission of the data, possibly accompanied by transmission of housekeeping information (such as destinations, audio or video trimming, and the like) is provided substantially continuously for an indeterminate period such as until termination of the connection 32. Although it may be that not every bit transferred represents a data bit (since "housekeeping" bits may also be transferred), the transfer of data is substantially continuous in the sense that there are no substantial periods during 35

which no data bits are transferred. It is possible that the data being transferred is "Null" data such as silence during a telephone conversation or transfer of a blank video image. One type of isochronous data transfer protocol is the Fiber Distributed Data Interface-II (FDDI-II) as described, for example, in *FDDI-II Hybrid Multiplexer*, Revision 2.4, dated Mar. 25, 1991.

Previous systems which were configured to use only a single-type protocol had the disadvantage that it was not possible to operate a mixed-protocol or "mixed-environment" system. Also when upgrading a network system, it was necessary to upgrade the entire system and it was infeasible or wasteful to upgrade only part of the system (such as only some of the nodes or such as upgrading nodes without upgrading hubs or upgrading hubs without upgrading nodes). Additionally, when a system or system components were installed, or repaired it was necessary for the installing personnel to be familiar with the particular single protocol for which the network was configured and to make such installation, upgrade, or repair in accordance with such a single protocol. Furthermore, it was necessary that apparatus connected to the system be configured for exclusive operation in accordance with the predetermined single protocol.

SUMMARY OF THE INVENTION

The present invention includes a recognition of the problems found in previous devices. According to an embodiment of the present invention an apparatus connected to one endpoint of a network link is able to detect which type of link signal, out of a number of possibilities is being received, thus indicating the protocol capability of the apparatus connected to the other end of the network link. In one embodiment, the apparatus is able to generate one of a plurality of link signals for transmission to the far end of the link, depending on the capabilities of each end of the link. Preferably, a first end of the network link has a capability of providing data communication under at least two different protocols and can select the appropriate protocol depending on what type of protocol capability is detected in the apparatus at the other end of the link.

Link endpoint capability detection takes advantage of the fact that different data communication protocols provide signals on the physical medium which have different characteristics. The various protocols can typically be detected by their unique timing and data patterns. According to one aspect of the invention, the network has a star topology with at least one hub and a plurality of nodes each node being connected to a hub by physical media constituting the link. The capability detection of the present invention can be performed by apparatus at either end of a link, and in particular, in a star topology network can be conducted by the hub or by any node. In one embodiment, capability detection is initiated by the hub. In a non-star topology at least one node can operate under two or more protocols and can detect the capability of another node with which it is connected.

Although, for convenience, much of the following description is in terms of hubs and nodes, aspects of the present invention can be implemented in topologies other than hub-and-node topologies (e.g., ring topologies, and tree topologies) as will be apparent to those of skill in the art. Descriptions of hub circuitry in the following can be implemented, e.g., on a PBX adapter card for a personal computer.

The apparatus which initiates capability detection, according to one embodiment, transmits a signal onto the

physical medium. In one embodiment, the apparatus at the far end of the link outputs, onto the physical medium, a second signal. Preferably, a second signal will be output from the apparatus at the far end of the link, regardless or whether the apparatus at the far end operates according to a first protocol or a second protocol. However, the second signal which is placed onto the physical medium at the far end of the link has either a first form or a second form, depending on whether the apparatus at the far end has a first protocol capability or a second protocol capability. This difference in signal is detected at the first end of the link and this could be used as a basis for determining the protocol capability at the far end of the link.

In another embodiment, the first apparatus outputs a first signal. The second apparatus outputs a response only if it has a first protocol capability. If no response is output, the first apparatus outputs a second signal in an attempt to elicit a response according to a second protocol. This process can be repeated until the first apparatus outputs a signal to which the second apparatus responds, thereby indicating a protocol capability of the second apparatus.

According to one embodiment, the first signal which is output, also carries information regarding the protocol capability of the first endpoint. That is, preferably, the first signal has a first form if the first endpoint has a first protocol capability and it has a second form if the first endpoint has a second protocol capability. Preferably, the apparatus at the far end of the link will respond to either of these forms in the manner described above.

In the preferred embodiment, the apparatus which has detected the capability at the far endpoint adjusts its operation to accommodate that capability. For example, when the first endpoint detects that the far endpoint has a first protocol capability, the first endpoint will configure itself to conduct subsequent communication using the first protocol. However, if the first endpoint detects that the far endpoint has a second protocol capability, the first endpoint is able to configure itself to accommodate the second protocol capability.

In one embodiment the far endpoint will have only a single protocol capability. However, it is possible to configure a network in which both link endpoints have multiple protocol capabilities and both can detect one or more capabilities at the opposite endpoint. The endpoints can then configure themselves to operate at the best or most desired protocol level.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. 1A, 1B and 1C of the timing of a packet transmission system, a token ring transmission system, and an isochronous transmission system respectively;

FIG. 2 is a schematic block diagram showing three nodes connecting to a hub;

FIG. 3 is a schematic block diagram showing a number of hubs connected together using a ring structure;

FIG. 4 is a schematic block diagram of circuitry for multiplexing and preparing data for transmission over to media and for receiving information from the media and demultiplexing the data;

FIG. 5 is a schematic block diagram of hub receiver circuitry according to an embodiment of the present invention;

FIG. 6 schematic block diagram of a hub transmitter-circuitry;

FIG. 7 in a timing diagram showing the relative timing of transmissions and receptions at the hub and nodes:

FIGS. 8A-8E are block diagrams depicting link endpoint capability detection for five different network configurations according to embodiments of the present invention:

FIG. 9 is a block diagram of a node receiver, according to an embodiment of the present invention:

FIGS. 10A, 10B and 10C depict timing of 10 Base T link pulses, isolink pulse stream and isosleep link pulses, respectively:

FIG. 11 depicts a state machine for generating various types of link pulses:

FIG. 12 depicts a state machine for detection of 10 Base T links:

FIG. 13 depicts a state machine for detection of isochronous or isochronous-ethernet pulses or fast link pulses:

FIG. 14 depicts pulses for defining acceptance windows: and

FIG. 15A-15D are block diagrams depicting circuitry components that can be used to implement state machines according to one embodiment of the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Before describing link endpoint capability detection, a general description of one type of network will be provided as one example of a data communication system in which the present invention can operate. A data communication system can be configured in a star-topology with a plurality of nodes of 42a, 42b, 42c, (FIG. 2) each coupled to a hub 44a by data links comprising physical data transmission media such as one-way twisted pair wires 46a-46f. The number of nodes can be adjusted depending on the data transmission needs and objectives of the system. In one embodiment, each hub is configured to accommodate connection with up to 16 nodes. Each 42a, 42b, 42c includes circuitry 50a, 50b, 50c for receiving data, converting it to a form suitable for transmission onto the physical media 45a, 46c, 46e using transmitters 78a, 78b, 78c and receipt of signals from the physical media 46b, 46d, 46f using receivers 79a, 79b, 79c and conversion to a form suitable for use by the data sinks. Each of the nodes 42a, 42b, 42c includes data sources and sinks 48a-48g. The data sources and sinks can be isochronous sources and sinks such as video cameras 48a, 48d and monitors 48b, 48a, non-isochronous sources and sinks such as an Ethernet media access controller 48c, 48g, and signaling or D channel sources and sinks such as an emulated or virtual key pad 48f provided, for example, on a personal computer (PC) terminal.

Each of the nodes 42a, 42b, 42c can include various types of sources and sinks such as strictly isochronous sources and sinks, such as depicted for node one 42a, strictly non-isochronous sources/sinks as depicted for node three 42c or both isochronous and non-isochronous sources and sinks as depicted for node two 42b. The physical layer 52 of the network system depicted in FIG. 2 has portions which include the physical media 46a-46f and physical layer devices such as the node data receivers and converters 50a, 50b, 50c and the hub components 54a, 54b, 54c and 56. The physical layer can include devices for providing, e.g., transceivers for 10 Base T cables, data multiplexing, phase locked loop circuitry, FIFOs or other circuitry for cable length adjustment, smoothing, squelch and the like.

The hub 44a includes circuitry 54a, 54b, 54c for receiving data from the physical media 46a, 46c 46e separating the

isochronous-sourced data from the non-isochronous-sourced data and the D channel and M channel data and converting separated data into a form suitable for handling by downstream hub circuitry 56. In the depicted embodiment the separated isochronous-sourced data is provided to a time slot interchange controller 58 for placing the data on a high-bandwidth bus (e.g. the TSI bus) so that it can be transported to destination nodes or other TSI controllers in the hub or other hubs (as depicted, e.g. in FIG. 3) and/or retrieved by hub circuitry 54a, 54b, 54c for transmission to various destination nodes 42a, 42b, 42c, e.g. by a local loopback capability. The separated non-isochronous data is provided to circuitry 60 configured to convey the non-isochronous data to one or more of the hub circuitry 54a, 54b, 54c for transmission to destination nodes 42a, 42b, 42c. In an embodiment in which non-isochronous-sourced data includes Ethernet data, the hub circuitry 60 can be a standard Ethernet repeater processor. In this way, the present invention can be at least partially backwards-compatible with previous Ethernet hub systems. The D channel and M channel information is provided to a signaling processor 62 which performs various maintenance and control functions such as identifying and alerting users of error conditions, and setting up requested calls or connections, i.e. source/destination paths e.g. by communicating with the isochronous and non-isochronous controllers 58, 60, e.g. over data path 64. The isochronous controller 58 can be, a data exchanger e.g., an isochronous switching device, such as that described in commonly-assigned application Ser. No. 08/146,337 entitled "Time Slot Exchanger Switching Mechanism in a Network for Data Communication Having Isochronous Capability," filed on even date herewith and incorporated herein by reference. The non-isochronous controller 60 can be, e.g., an Ethernet repeater. As depicted in FIG. 3 a number of star-topology systems can be interconnected by connecting hubs 44a, 44b, 44c to one another, e.g. by a time slot interchange (TSI) ring.

According to the present invention, data communication can be provided according to one or more of a number of protocols. Those skilled in the art are familiar with protocols, but in general, a "protocol" includes a standard set of rules that specify the format, timing, sequencing and/or error checking for data transmission. Several network protocols are referenced above, including an Ethernet protocol such as 10 Base T, an isochronous protocol such as FDDI-II, and a token ring protocol. Another possible protocol is one in which both isochronous and non-isochronous data are combined into a frame structure for transmission across physical media. A frame-structure protocol of this type is described in greater detail in commonly-assigned application Ser. No. 07/969,916 titled "Network for Data Communication with Isochronous Capability" filed on Nov. 2, 1992 and incorporated herein by reference. According to one such protocol, the incoming data from the various sources is provided to a multiplexer 70 (FIG. 4) which performs time-division multiplexing on a four-bit basis. The pattern for the time division multiplexing is a repeating series of frames or templates. In this embodiment, the frames are repeated every 125 microseconds. The time division multiplexing is a multiplexing of isochronous-sourced data and non-isochronous-sourced data. The non-isochronous-sourced data can be data provided in accordance with a number of previously-available LAN systems and this protocol will be referred to, in general, as "isochronous-LAN" protocol. Several particular types of isochronous-LAN protocols are possible. When the isochronous data is multiplexed with LAN data which is provided according to an Ethernet protocol, such as a 10

Base T Ethernet protocol, the resulting time-division multiplexed protocol will be referred to as an "Isochronous-Ethernet" protocol. When the isochronous data is multiplexed with LAN data which is provided according to a token ring protocol, the resultant time multiplexed protocol will be referred to as an "isochronous-token ring" protocol.

The present invention will be described below by way of a particular example in which one available protocol is an Isochronous-Ethernet protocol and another potentially available protocol is a 10 Base T protocol. However, as will be clear to those skilled in the art, the present invention can also be used in connection with other combinations of protocols such as isochronous-token ring or other isochronous-LAN protocols, pure isochronous protocols such as FDDI-II, and can include three or more protocols.

Tables IA and IB depict manners in which the various data streams, and additional data and control bytes can be time-division multiplexed in an Isochronous-Ethernet protocol. Each symbol in the Tables IA and IB represent four bits of data so that every group of two symbols represents one 8-bit byte of data. In Table IA, E represents four bits of data from the non-isochronous Ethernet stream 66b (FIG. 4), B designates four bits of data from the isochronous stream 66a, D represents four bits of data from the signaling or D channel stream 66c, and M represents four bits of M channel data stream 66d. In addition, certain byte-length patterns are provided. JK represents a frame synchronization pattern and EM (the first two bytes of block three in Table IA represents an Ethernet "pad" followed by a maintenance byte. As seen in Table IA each frame contains 256 bytes which can be considered in thirty-two groups of eight bytes each, or four blocks of sixty-four bytes each. The frame structure is described more thoroughly in commonly-assigned application Ser. No. 07/969,911 titled "Network for Transmitting Isochronous-Source Data with a Frame Structure" filed Nov. 2, 1992 and incorporated herein by reference. Frame structures other than that described in Table IA may be used to allocate bandwidth according to a particular purpose. Table IB shows one of the many alternate formats. In general, Table IB is similar to Table IA with replacement of "E" symbols with "B" symbols. As seen in Table IB, the last one or two bytes in each block are "Idle" data bytes.

As shown in FIG. 4, the time-multiplexed data is then encoded by an encoder 72. In the depicted embodiment, the encoder performs four/five encoding. One particular form of four/five encoding conforming partially to the ANSI X3T9.5 standard, is depicted in Table II. The encoding scheme depicted in Table II is described in greater detail in commonly-assigned application Ser. No. 07/970,329 titled "Frame-Based Transmission of Data" filed on Nov. 2, 1992 and incorporated herein by reference.

The output from the encoding devices is sent to pre-emphasis circuitry 76. The pre-emphasis circuitry compensates the signal transmitted onto the physical medium to reduce the jitter. The data output by the pre-emphasis circuitry 76 is sent to a transmitter or driver 78b and the signal is transmitted over the physical medium 46c. The physical medium 46c can be any of a number of media types including twisted pair, coaxial or fiber optic cable.

The data sent over the physical layer interface is received in the hub 44a. The hub contains a plurality of circuit devices 54a, 54b, 54c, each one coupled to one of the nodes 42a, 42b, 42c by the physical media 46. As depicted in FIG. 5 the data transmitted over the physical media 46 arrives serially at a de-serializer/decoder 80. Link detect circuitry 82 also receives the data from the physical layer portion inter-

face for detection of the mode or protocol in which the node is operating as described more fully below. The de-serializer/decoder 80 receives a reference clock signal 84. The de-serializer/decoder includes circuitry which is functionally an inverse of the multiplexing/encoding circuitry described above. Referring to FIG. 4, the de-serializer/decoder includes phase lock decode circuitry 86, the results of which are provided to NRZI decode circuitry 88 which, in turn, provides the decode results to four/five decode circuitry 90, in turn providing results to a de-multiplexer 92 which separates the received data into the isochronous-sourced data 94a, the non-isochronous-sourced data 94b and signaling data, such as D channel and M channel data 94c. The de-serializer/decoder 80 also outputs a synchronization signal, derived from the JK frame synchronization symbols 96 for use by a framing timing generator 98.

Both the non-isochronous-sourced data 104 (FIG. 5) and the isochronous-sourced data 102 are made available to the various hub circuitry 56 or components 54a, 54b, 54c, as needed for transmission back to destination nodes. In one embodiment, the separated isochronous data 102 and non-isochronous data 104 are reconfigured by the respective interfaces 58, 60 to provide isochronous output 102 and non-isochronous output 104 in a form suitable for processing so as to provide the data as needed for transmission to the destination nodes via the physical layer. In one embodiment, the non-isochronous data 94b can be configured by the physical layer E interface 59 (FIG. 5) so that the output data 104 can be processed by a repeater device 60 for eventual transmission to destination nodes. As an alternative to using a repeater for the non-isochronous data, packet connections may be linked through media access control layer bridges. Preferably, the output data 104 is in a form such that it can be handled by repeater circuitry of types previously available. For example, when the nonisochronous data 94b within the physical layer is data which originated at the node 42b from an Ethernet MAC, the output data 104 is in a form such that it can be handled by a standard Ethernet hub repeater 60 such as DP83950 "Repeater Interface Controller" (RIC) available from National Semiconductor Corporation, Santa Clara, Calif.

As shown in FIG. 5, the data received over the physical link 46 is also provided to an additional interface for handling data according to a second protocol, as described more thoroughly below. For example, when the second protocol is an Ethernet 10 Base T protocol, a 10 Base T interface 512 can be provided. The 10 Base T receive interface 512 can be a standard 10 Base T interface, such as Model DP83922 "Twisted Pair Transceiver Interface" (TPI) available from National Semiconductor Corporation, Santa Clara, Calif. A multiplexer 514 determines whether the repeater 60 receives a data stream from interface 59 or the 10 Base T interface 512. This selection by the multiplexer 514 is controlled by a mode select signal output over control line 516 from the link beat detect circuit 82 as described more fully below.

The data 198 output from the E transmit interface 168 is provided along with isochronous data output 164 and M channel D channel data 170 to encoder serializer circuitry 202, depicted in FIG. 6. The encoder/serializer 202 is configured substantially like the encoding circuitry found in the node and depicted in FIG. 4. Specifically, the encoder/serializer 202 provides a multiplexer for combining the three streams of data 198, 170, 164, a four/five encoder, an NRZI encoder, and pre-emphasis circuitry. The timing of transmission is controlled by transmit timing circuitry 204. Output 206 from the encoder/serializer is selectively com-

bined with link beats from a link beat generator 208 by
 multiplexer 210 for purposes of link end point detection, as
 described below. The clock signal and the data 166 from the
 repeater 60, in addition to being provided to the E interface
 168 is also provided to a second interface which operates
 according to a second protocol. When a second protocol is
 an Ethernet 10 Base T protocol, the interface is an Ethernet
 10 Base T interface 520. The Ethernet 10 Base T interface
 transmit 520 can be of a type substantially identical to 10
 Base T interfaces provided in previous apparatus such as
 Model DP83922, "Twisted Pair Transceiver Interface (TPI)"
 available from National Semiconductor Corporation, Santa
 Clara, Calif. The output from the Ethernet 10 Base T
 interface 520 is provided to the multiplexer 210. Multiplexer
 210 is able to select, in response to a control signal 522,
 whether to output data originating from the repeater 60
 according to a first protocol determined by the E interface
 168, or according to a second protocol determined by the
 Ethernet 10 Base T interface 520, as described more fully
 below. The data sent from the hub 44a to the nodes 42 is sent
 in a frame format which is preferably substantially the same
 as the frame format used for the data sent from the nodes 42
 to the hub 44a as described above. At the nodes 42, the
 circuitry 50 includes devices (FIG. 4) for decoding and
 de-multiplexing data, similar to that described for perform-
 ing these functions in the hub, mainly a phase lock decode
 86, and NRZI decode 88, a four/five decode 90, and a
 de-multiplexer 92. Decoded and de-multiplexed data is then
 delivered to the various data sinks in the nodes 42.

As shown in FIG. 7, the timing of the system can be
 synchronized with a 125 microsecond reference clock signal
 214. In this example, the reference signal 214 provides an
 ascending clock edge every 125 microseconds. The refer-
 ence signal can be provided by any of a number of sources.
 Preferably, an embodiment of the present invention is con-
 figured to permit a reference signal 214 to be synchronized
 to an external clock reference, such as a reference signal
 from a wide area network or from a FDDI-II ring. The
 reference signal can be supplied through one of the nodes
 and transmitted to the hub for distribution to the other nodes,
 or can be supplied directly to the hub for distribution.

FIG. 8A depicts a network configuration in which the hub
 530a is a 10 Base T hub and the node 532a is a 10 Base T
 node, both of which are found in previously-available
 devices. In this system, the 10 Base T hub sends a signal,
 specifically a link test pulse, in accordance with IEEE
 Standard 802.3, over the physical medium to a 10 Base T
 node 532a. In a typical system, the 10 Base T hub outputs
 a signal upon being powered-up. The link test pulse used in
 previous devices is described in IEEE Standard 802.3.
 Briefly, a link test pulse can be described as a single 100
 nanosecond pulse occurring at a nominal 16 millisecond
 interval. The 10 Base T node 532a, typically in response to
 being powered-up, outputs onto the physical medium a
 second signal, which, in accordance with IEEE 802.3, is
 substantially identical to the above-described link test pulse.
 This link test pulse is received by the 10 Base T hub 530a.
 At this point, a 10 Base T hub proceeds to operate on the
 basis that it is connected to a 10 Base T node (532a) and the
 node 532a begins to operate on the basis that it is connected
 to a 10 Base T hub (530a) and normal 10 Base T commu-
 nication proceeds.

FIG. 8B depicts a configuration according to one inven-
 tion in which an Isochronous-Ethernet hub 530b is con-
 nected to a 10 Base T node 532a. The Isochronous-Ethernet
 hub outputs a probe signal 534. A probe signal differs from
 the link test pulse in that it has a faster link beat, for example

having a beat period of less than about 2 milliseconds. The 10 Base T node 532a is configured substantially identically to previously available 10 Base T nodes. Upon receipt of the probe pulse 534, it continues to output a link test pulse onto the physical medium. The isochronous-Ethernet hub 530b, upon receiving a link test pulse (rather than a probe pulse) can determine, on that basis, that the apparatus connected to the far end of the physical medium is a 10 Base T node 532a (rather than, for example, an isochronous-Ethernet node). Preferably, the isochronous-Ethernet hub 530b is capable of handling data either according to an isochronous-Ethernet protocol or a 10 Base T protocol. Upon receiving a link test pulse and determining that the node 532a is a 10 Base T node, the isochronous-Ethernet hub 530b will configure itself to conduct all future communications with node 532a using a 10 Base T protocol.

Although FIG. 8B shows only a single node 532a connected to the hub 530b, in a typical configuration, a plurality of nodes will be connected to each hub. Preferably, the hub 530b is capable of using different protocols with different nodes. Accordingly, an isochronous-Ethernet hub which is connected to both a 10 Base T node and an isochronous-Ethernet node can determine the capability of each node to which it is connected and can use the appropriate protocol for each node.

FIG. 8C depicts a network configuration in which a 10 Base T hub 530a is connected to an isochronous-Ethernet node 532b. Upon initialization of the system the 10 Base-T hub outputs a link test pulse 533. In the depicted embodiment the Isochronous-Ethernet node 532b can operate according to an Isochronous-Ethernet protocol. Therefore, upon receiving the link test pulse 533, it outputs a link test pulse 533. Accordingly, the 10 Base T hub 530a can only send Ethernet data and no isochronous data.

FIG. 8D depicts a network configuration in which an Isochronous-Ethernet hub 530c is connected to an isochronous node 532c. In the embodiment depicted in FIG. 8D, the node 532c only has isochronous protocol capability, but the hub 530c has both an isochronous-Ethernet protocol capability and an isochronous protocol capability. In this embodiment, upon initialization of the system, the hub 530c outputs an isochronous probe pulse 535. The isochronous node 532c, upon receiving the isochronous probe pulse 535, can determine that the hub to which it is attached is an isochronous-capability hub and will configure itself to conduct all future communications with the hub 530c according to an isochronous protocol. Thus, the isochronous node 532c preferably contains isochronous apparatus similar to apparatus found in the hub 503c for detecting circuitry at the other end of the link or physical medium and, thereafter, using the appropriate protocol. The isochronous node 532c, in response to receipt of the iso probe pulse 535, outputs an iso probe pulse 535. The hub 530c, upon receipt of the iso probe pulse, will commence normal isochronous hub operations.

FIG. 8E a configuration in which an Isochronous-Ethernet hub 530b is connected to an Isochronous-Ethernet node 532b. When the system is initialized the Isochronous-Ethernet hub 530b outputs a probe signal such as an isoEnet probe, on the physical medium. When the Isochronous-Ethernet node 532b receives the isoEnet probe signal it is able to determine that the hub to which it is connected is an Isochronous-Ethernet hub. The Isochronous-Ethernet node 532b then outputs an isoEnet probe signal 534 onto the physical medium which is received by the isochronous Ethernet hub 530b. When the Isochronous-Ethernet hub 530b receives an isoEnet probe signal it can determine that

the node 532b to which it is connected is an Isochronous-Ethernet node and will conduct all future communications with this particular node according to the Isochronous-Ethernet protocol.

FIGS. 5 and 6 depict components in the hub which are used in connection with link endpoint capability detection. As depicted in FIG. 6, a link beat generator 208 is provided for outputting the appropriate probe signal. A control signal 522 controls the multiplexer 210 so that the probe signal 208 is output onto the physical medium 46 at the appropriate time, e.g., upon initialization of the network system.

FIG. 9 depicts circuitry 50 in an Isochronous-Ethernet node. Circuitry 542 would be provided in a node which can operate in accordance with two protocols. Nodes which provide only a single protocol would not include circuitry 542 and the E interface 59' would be connected directly to the Ethernet MAC 49c. The probe pulse received over the physical medium 46 is detected by the link beat detector circuitry 82. Circuitry 82 can include, for example a state machine, for detecting the sequence and interval of the pulse or link test pulse. The link beat detector circuitry 82 outputs a mode select signal 516 for controlling the multiplexer 514. The control signal 516 is configured to set the multiplexer 514 such that the Ethernet MAC 48C is connected to the output of the Isochronous-Ethernet interface 59' so that future data received over the physical medium 46 is treated in accordance with the Isochronous-Ethernet protocol. If the link beat detector 82 detects the link test pulse rather than an iso pulse signal, it outputs a mode select signal 516 which configures the multiplexer 514 to connect the Ethernet MAC 48C with the Ethernet 10 Base T interface 512 so that future data received over the physical medium is treated in accordance with Ethernet 10 Base T protocol. The mode select signal 516 also provides a signal to a control circuit in a node transmitter. The node transmitter is not separately depicted in detail since it is substantially identical to the hub transmitter depicted in FIG. 6.

The node transmitter control 522 in response to the mode select signal 516 (indicating receipt of a link test pulse or other probe pulse) configures the multiplexer to output an appropriate pulse signal from the link beat generator 208 onto the medium 46. In some embodiments, nodes and/or hubs are configured to output a link test pulse or a probe pulse (depending on the capability of the hub or node), whenever the hub or node is powered-up. For embodiments in which the link beat detect 82 is able to discriminate between a link test pulse and a probe signal such as an iso probe pulse, the mode select 516 can configure the link beat generator 208 to output a link test pulse in response to a link test pulse and an iso probe pulse in response to a probe signal. The signal output by the node transmitter is received in the hub receiver 54 (FIG. 5). The hub receiver link beat detect circuitry 82 detects the output of the probe pulse from the node transmitter. When the signal is a probe signal, circuitry 82 outputs a mode select signal 516 which is effective to control the multiplexer 514 to connect the output from the E interface 59' to the repeater 60. In this way, the hub receiver is now configured to process future signals received from the node over medium 46 according to an Isochronous-Ethernet protocol. The node select signal 516 also provides an input to control signal 522 which, in response, configures the multiplexer to place the output 206 from the encoder/serializer 202 onto the physical medium 46, rather than using the output from the 10 Base T interface 536. In this way, the transmitter is now configured to output data according to the Isochronous-Ethernet protocol.

If the signal output from the node is a link test pulse rather than probe pulse, the link beat detector 82 outputs a mode

select signal 516 which configures multiplexer 514 to connect the Ethernet 10 Base T interface 512 with repeater 60 and configures the multiplexer to send output 536 onto the physical medium 46, rather than output 206.

In one embodiment, generation and detection of link pulses involves a number of changes of state, as described below by way of state machine descriptions and diagrams. In one embodiment, the operation can be described by three state machines, a first state machine for generating various types of link pulses ("LINKGEN"), a second state machine for detection of a 10-base T link ("LINK10BTSM") and a state machine for detection of isochronous or Isochronous-Ethernet pulses or fast link pulses ("LINKISOSM"). 10 Base T link pulses are transmitted and, in turn, detected on both sides of the medium such as the twisted pair medium, to signal the proper connectivity. In isochronous systems, the fast link pulses are generated during power-on initialization, during traumatic error recovery, or when a connection is running on emergency power. Fast link pulses can be differentiated from 10 Base T link pulses since the fast link pulses occur in bursts rather than singly. A third type of link pulse "isosleep" is used to indicate that the device originating the pulses is in a low power or "sleep" mode and to convey cycle timing. Low power mode is described in commonly assigned application, U.S. Ser. No. 08/147,359 for "Low Power Isochronous Networking Mode" filed on even date herewith and incorporated herein by reference. The 10 Base T link pulses have the form of a 100 ns pulse generated every 16 ms (FIG. 10A). In the depicted embodiment, the isolink pulse stream consists of pulse pairs. Each pair consists of a clock pulse and a data link pulse. In the depicted embodiment, the spacing between the clock pulse is 125 μ s. This value is preferred because it is the same as the public network time and it is a clock time that is readily available to the system, as described above. The clock and data link pulses are separated from each other by 62.5 microseconds. The pairs are repeated 16 times and, following the 16th transmission of a pulse pair, an additional link pulse 1006 is transmitted 62.5 microseconds after the last data link pulse position. The isolink pulse stream is depicted in FIG. 10B. As shown, clock link pulses 1002a, 1002b always occur, while data link pulses 1004a, 1004b occur to represent a data "1" (shown in phantom) and are missing to represent a data "0". Thus, the isolink pulse stream can be used to transmit information and, in one embodiment, is used to encode information such as the type of device which is transmitting, (e.g., hub versus node) the isoethernet signaling data rate, and the information content of the isoethernet channel (e.g., clear channel, ATM mode, isochronous Ethernet).

The isosleep link pulses consist of one pulse 1020a, 1020b transmitted every 125 μ s in phase with the transmit sync signal, as depicted in FIG. 10C.

In one embodiment, the hub initially begins generating an isoethernet "fast" link pulse to each node to which it is connected. If the far end is a 10 Base T node, this node will begin transmitting a 10 Base T link pulse after it has received the pulse or pulse train sent from the hub. If a 10 Base T node at the far end fails to receive a proper link pulse or stream of link pulses, it will enter a "link loss" state in which it will remain until it receives a specific sequence indicating that the network or link is now operable again. When the hub receives a 10 Base T link pulse from the node, it will configure itself to thereafter send out 10 Base T communications to that node.

If the far end of the link was an isoethernet node, the isoethernet node will respond to receipt of a proper isoeth-

ernet pulse train (fast link) by transmitting an isoethernet pulse train (fast link). Thereafter, both ends of the link will configure themselves to transmit in isoethernet mode.

In this way, the hub will be assured that the communication link is working properly in both directions. In certain previous systems, communications did not require a "handshake," i.e., verification of properly working link in both directions and accordingly, in these previous devices it was possible for there to be a partially broken link (e.g., a link which was operating in one direction and not the other) that went undetected.

FIG. 11 depicts the LINKGEN state machine according to one embodiment of the invention. As seen in FIG. 11, this state machine has four states. State LG1 "LINKIDLE" is an idle state in which the link generator state machine is waiting for an event to generate a link pulse. State LG2 1104 is a link pulse state. The machine waits at this state for approximately 250 ns until the end of the interval when the cycle synchronization symbol is sent before generating a data link pulse. State LG3 1106 is the link data state. The machine waits at this state to generate a data link pulse. State LG4 1108 is the link clock state. The machine waits at this state to generate a clock link pulse. Equations governing the transition between states in pseudo "C" terminology are provided in Table III. In each case, the first line indicates the condition which causes the transition of state to occur and the remaining lines indicate values that are assigned, incremented, cleared or reset. The meanings of terms in Table III are provided in Table IV.

As seen from FIG. 11, the state machine leaves the link idle state under three conditions. The first 1114 is a 10 Base T link pulse timeout designated LG11. After waiting 16 ms from the previous link pulse, a link pulse is generated. In the isosleep link pulse timeout designated LG12 (1112), after waiting 125 μ s from the previous link pulse, a link pulse is generated. In the isolink link pulse timeout designated LC13 (1110), after waiting 16 ms from the previous link pulse, a link pulse stream will be generated.

The state machine leaves the link pulse state 1104 after generating a link pulse. In the isolink data wait, after generating a link pulse, the machine makes a transition to begin timing the data link pulse. In the isolink clock wait, after generating a data link pulse, the machine makes a transition to begin timing the clock link pulse.

The machine leaves the link data state in either of two conditions. In the isolink 1 data pulse, after waiting a half cycle, a data link pulse will be transmitted. In the case of an isolink 0 data pulse, after waiting a half cycle, no data link pulse will be transmitted.

The machine leaves the link clock state 1108 in the case of an isolink clock pulse. After waiting a half cycle, a clock link pulse will be transmitted.

The 10 Base T link detection state machine ("LINK10BTSM") is depicted in FIG. 12. This state machine can be compared to the 10-Base T detector described in IEEE Standard 802.3. However, the state machine depicted in FIG. 12 includes the state "freeze-10-Base" 1202 which acts to disable this machine when fast link pulses are detected. Table V indicates the conditions for the various transitions shown in FIG. 12 and the variable assignments that are associated with each transition.

Table VI indicates the meaning of various parameters. Following a reset 1204, the machine will enter the link test reset state 1208. From this state, the machine will either remain in this state 1210, transition to the link test fail state 1212, transition to the link test extend state 1214 or transi-

tion to the freeze-10-base state 1202. The transition to the freeze-10-base state occurs if the fastlink parameter is "true". The same conditions will also cause a transition from the link test fail state 1212 or the link test extend state to the freeze-10-Base state. Once in the freeze-10-Base state 1202, the state machine will, by default, remain in this state 1202 as long as the fastlink parameter is "true." In this situation the freeze-10-base state will transition to the link test reset state 1208. In this way, the state machine will respond to receipt of a normal 10-Base T link pulse but will enter the freeze state 1202 in response to receipt of a fast link pulse.

The state machine which detects a fast link pulse ("LINKISOSM") is depicted in FIG. 13. Table VII indicates the conditions which cause state transitions as well as the assignment of variables or parameters associated with state transitions. Table VIII indicates the meaning of the various parameters.

The state machine depicted in FIG. 13 has nine separate states. The iso-idle state 1302 is entered upon reset or receive activity. The isofirst pulse state 1304 is entered when a link pulse is detected but at a point when it does not yet qualify as a fast link pulse. The isofirst pass state 1306 is entered when a qualified fast link pulse is detected. The isosecond pass 1308 state is entered when a qualified second link pulse is detected. The isothird pass state 1310 is entered when a qualified third link pulse is detected. The isofast link state 1312 is entered after the requisite number of fast link pulses has been detected, but when the machine is waiting for data information or a clock pulse. The isofind clock state 1314 is entered when a clock pulse within a burst is detected. The isoshift 0 state 1316 is entered when 0 data pulse is detected and the isoshift 1 state 1318 is entered when a 1 data pulse is detected.

To distinguish between a clock pulse and a data pulse, a series of acceptance windows are defined from the beginning of the first pulse which is assumed to be a clock pulse. As depicted in FIG. 14, at the beginning of the first clock pulse 1402, a first acceptable clock window 1404 of 31.25 μ s is defined. Thereafter, a series of acceptance windows, alternating between an acceptable data window 1406 and an acceptable clock window 1408 are defined, each of these windows having a duration of 62.5 μ s.

The state machines, described above, can be implemented in the context of a number of circuitry components. In one embodiment, the circuitry components include a link timer, FIG. 15A, a number of link registers 1502 (FIG. 15B), coupled to a link comparator 1504, the link ISO state machine (FIG. 15C) and the link 10 Base T state machine (FIG. 15D). Table IX describes the function of the various signal lines depicted in FIG. 15A-15D, not previously described.

In general, the link timer circuit 1506 provides a number of timers which are used by the state machines to distinguish between pulse signals and other signals and to distinguish between various types of pulses and pulse streams, as described above. A number of the timers found in these circuits, and the function and default values, are listed in Table X.

The link registers 1502 are used for storing information, including information encoded in the data pulses of the isoethernet pulse stream and for outputting information, such as information extracted from the data pulses.

In view of the above description, a number of advantages of the present invention can be seen. The present invention allows a network to be configured in a mixed protocol or mixed environment, with, for example, a single hub con-

nected to a plurality of nodes which operate according to different protocols, with the configuration being achieved automatically, without the need for manually establishing a predetermined protocol beforehand for each node. The present invention permits networks to be upgraded incrementally so that it is not necessary to upgrade all nodes at the same time. Furthermore, it is not, in general, necessary for service personnel to specifically configure nodes or hubs to accommodate particular protocols since the protocols are determined automatically and the nodes and hub configure themselves in accordance with the determined protocols.

A number of variations and modifications of the present invention can be used. Although an embodiment involving a 10 Base T protocol and an Isochronous-Ethernet protocol was described, the present invention can be made applicable to other protocols, including other LAN protocols such as a token ring protocol, an isochronous protocol and the like. Although the present invention described one particular signal characteristic used for determining the protocol, other characteristics could also be used. For example, a token ring could be detected by the presence of four or 16 Mbit/sec Manchester-encoded data. Other LANs can be detected by their unique timing and data patterns. Protocols could also be detected using such characteristics as the pattern of the presence or absence of a carrier, and the frequency spectrum of signals placed onto the physical medium. When a node has a capability of communicating under two or more protocols, e.g. either an Isochronous-Ethernet protocol or a pure Ethernet protocol, it would be possible for a hub to use both capabilities of a node, i.e., to communicate according to a first protocol during a first time period and a second protocol during a second time period. Although the present invention has been described in the context of a star topology, the invention could also be used in a non-star topology, such as a ring topology or a tree topology. The present invention can be used in networks which do not have a hub, such as direct connections between two nodes with each node determining the protocol capabilities of the other node. As described above, the link test pulse and iso probe signals are related in that, for example, a 10 Base T node will respond in the same fashion to receipt of either type of pulse. However, the test signals could be provided in forms which are unique to each type of protocol. In such a system, a data source/sink would output a first type of test pulse or other signal and, if no response was received, would output a second type of test pulse or signal, and so forth until a response was received indicating the protocol capability at the other end of the link. A data source/sink could be configured to determine all possible protocol capabilities of the apparatus at the other end of the link, rather than determining the "highest" or "best" capability available or using the first capability detected. The devices at each end could select a protocol capability other than the "highest" or "best" capability. It would be possible for a node to store an indication of its capabilities, such as in a table or other memory device, and to output the information upon receiving an inquiry. It would also be possible for a network to initialize in a common protocol, e.g., a 10 Base T protocol, and, thereafter, exchange information, using that protocol, indicating additional protocol capabilities of the components of the system. Thereafter, the systems could reconfigure themselves to use desired ones of the available protocols.

Although the present invention has been described by way of preferred embodiments and certain variations and modifications, other variations and modifications can also be used, the invention being defined by the following claims.

TABLE IA

[illegible][illegible][illegible][illegible]

JK = Frame Synchronization Pattern
MM = 8 Maintenance Bits
D = D Channel
EM = Ethernet Pad & 4 Maintenance Bits
E = Ethernet Packet Channel
8 = Isochronous Channel

TABLE IB

[illegible][illegible][illegible]

TABLE IB-continued

B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group22
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group23
BLOCK 3:																		
I	M	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group24
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group25
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group26
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group27
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group28
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group29
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	Group30
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	I	I	Group31

JK = Frame Synchronization Pattern

M = Maintenance Channel

D = D Channel

I = Idle Data

B = Isochronous Channel

TABLE II

20

Symbol	Encoded (5 bit)	Description	
0	11110	Data 0	
1	01001	Data 1	
2	10100	Data 2	
3	10101	Data 3	25
4	01010	Data 4	
5	01011	Data 5	
6	01110	Data 6	
7	01111	Data 7	
8	10010	Data 8	
9	10011	Data 9	30
A	10110	Data A	
B	10111	Data B	
C	11010	Data C	
D	11011	Data D	
E	11100	Data E	
F	11101	Data F	35
I	11111	No Ethernet Carrier	
S	11001	No Ethernet Data	
V	01100	Unaligned Data	
T	01101	Unassigned	
J	11000	Frame Sync Part 1	
K	10001	Frame Sync Part 2	40
Q	00000	Invalid	
H	00100	Invalid	
R	00111	Invalid	
V	00001	Invalid	
V	00010	Invalid	
V	00011	Invalid	45
V	00101	Invalid	
V	00110	Invalid	
V	01000	Invalid	
V	10000	Invalid	

50

TABLE III

LG 13:	isolink & 16ms	
	clr 16ms	
	index = 0	
	clr 62.5us	55
	lp = 1	
LG 11:	10baseT & (16 ms txcn)	
	clr 16ms	
	lp=txcn	
LG 12:	isoleep & sync	
	clr 16ms	60
LG 21:	!sync	
	lp = 1	
LG 34:	62.5 μ s	
	clr 62.5	
	lp=data[index]	65

TABLE III-continued

20	LG -13:	62.5 μ s & (index!=15) lp=1 inc index
25	LG -11:	62.5 μ s & (index==15) lp=1

TABLE IV

30	Term	Meaning	Value
	isolink	Enables Generation of isolink pattern	Not Applicable
	16ms	16 ms timer	16 ms
	62.5us	62.5 μ s timer	62.5 μ s
	lp	link pulse	Not Applicable
35	txen	Transmit enable	Not Applicable
	isosleep	Enables Generation of link low power link pulses	Not Applicable
	sync	Signal from template generator indicating start of cycle	Not Applicable
40	10 Base T	Enable. Generation of 10 Base T link pulses	Not Applicable

TABLE V

45	1 reset
	10baset = 1
	linkgood = 1
	setlht = 1
	setlmt = 1
50	bp1 = receive - (linkpulse * lmtDone)
	10baset = 1
	linkgood = 1
	setlht = 1
	setlmt = 1
55	bp2 = fastlink
	bp3 = lldone * receive * linkpulse
	loadic = 1
	10baset = 1
	bp4 = receive
	10baset = 1
60	bp5 = fastlink
	bp6 = receive * linkpulse
	10baset = 1
	setlmt = 1
	setltxi = 1
	bp7 = LCDone + receive
65	10baset = 1
	bp8 = fastlink

TABLE V-continued

bp9 = linkpulse * lmtDone lObaset = 1 decic = 1	5
bp10 = receive * ltxtDone - linkpulse * lmtDone lObaset = 1 loadlc = 1	
bp11 = receive * txc lObaset = 1 linkgood = 1 setlIt = 1 setlmt = 1	10
bp12 = fastlink * receive * txc	
bp13 = receive lObaset = 1	15
bp14 = receive lObaset = 1	
bp15 = fastlink loadlc = 1 lObaset = 1	20

TABLE VI

Term	Meaning	Value	
lObaset	Enable lObase-T Mode	Not Applicable	25
linkgood	lObaset state machine has detected good link pulses	Not Applicable	
setlIt	Set Limit Loss Timer	Not Applicable	
setlmt	Set Link Timer Min Time	Not Applicable	
receive	Now Called ipact	Not Applicable	
linkpulses	Link Pulses from Front End	Not Applicable	30
lmtDone	Link Time Min Timer Done	4 ms	
fastlink	lso Link Detector	Not Applicable	
lItDone	Link Loss Timer Done	60 ms	
loadlc	Load Link Counter Counts Link Pulses	Not Applicable	
setltx	Set Link Test Max Timer - Same as sltx	Not Applicable	35
LCDone	Link Counter Done	Not Applicable	
decic	Decrement Link Counter	Not Applicable	
ltxDone	Link Test Max Timer Done	32 ms	
txe	Transmit Enable	Not Applicable	
txc	Transmit Clock	Not Applicable	40

TABLE VII

bp1 = linkpulse setlmt = 1	45
bp2 = flmtDone	
bp3 = linkpulse * flmtDone setlmt = 1	
bp4 = flmtDone	
bp5 = linkpulse * flmtDone setlmt = 1	50
bp6 = flmtDone	
bp7 = linkpulse * flmt + Done setlmt = 1 fastlink = 1	
bp8 = flmtDone * linkpulse fastlink = 1 setlmt = 1	55
bp9 = flmtDone fastlink = 1 setltx = 1	
bp10 = receive setltx = 1 fastlink = 1	60
bp11 = receive * ltxDone reseureg = 1	
bp12 = linkpulse setlIt = 1 setlmt = 1 setltx = 1 fastlink = 1	65

TABLE VII-continued

5	bp13 = lmtDone
	fastlink = 1
	shiftrow = 1
10	bp14 = linkpulse * fdtDone
	shift1 = 1
	setlxt = 1
	setlmt = 1
	fastlink = 1
15	bp15 = linkpulse * fdtDone
	setlxt = 1
	setlmt = 1
	setldt = 1
	fastlink = 1
20	bp17 = linkpulse * fdtDone
	setlxt = 1
	setlmt = 1
	setldt = 1
	fastlink = 1
25	bp18 = lmtDone
	fastlink = 1
	shiftrow = 1
	unconditional
	setldt = 1
30	setlmt = 1
	setlmt = 1
	setlxt = 1
35	fastlink = 1

TABLE VIII

	Term	Meaning	Value
30	setlmt	Set Fast Link Max Timer	156.25 μ s
	fmdtDone	Fast Link Max Timer Done	Not Applicable
	resetreg	Reset Registers - Used to clear registers upon loss of carrier	Not Applicable
35	setldt	Set Fast Link Data Timer	93.75 μ s
	shiftrow	Shift Pointer to next row of link data shift register	Not Applicable
	fdtDone	Fast Link Data Timer Done	Not Applicable
	shift1	Shift a 1 bit into the link data shift register	Not Applicable
	shift0	Shift a 0 bit into the link data shift register	Not Applicable
40			

TABLE IX

	Signal	Meaning
45	ethclock	Ethernet clock
	farack	Far acknowledge - used in link protocol
	isogood	Used to indicate isolink pattern good
50	isotype	field in isolink pattern indicating ISO usage - normal, wideband, none, reserved
	loadrxmode	Load receive mode register
	porttype	Field in isolink pattern indicating port type - node, hub
55	reset	Hardware reset
	resetlink	Reset signal used to clear contents of the receive link registers
	rxmodebus	Conveys the contents of the received link pattern
60	setackrx	Set receive link acknowledge bit
	setlim	Set link initialization match - used to indicate that the link pattern was received consistently 3 times and was compatible with mode of device
	setlinm	Set link initialization no match - used to indicate that the link pattern was received consistently 3 times and was not compatible with mode of device
65		